

WEST Search History

DATE: Friday, September 29, 2006

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L44	downer-rober-victor.in.	0
<input type="checkbox"/>	L43	dietel-john-david.in.	10
<input type="checkbox"/>	L42	bestgen-robert-joseph.in.	16
<input type="checkbox"/>	L41	L36 and l24	0
<input type="checkbox"/>	L40	L35 and l24	0
<input type="checkbox"/>	L39	(l33 or l34) and l27	1
<input type="checkbox"/>	L38	(l33 or l34) and l26	1
<input type="checkbox"/>	L37	(l33 or l34) and l25	81
<input type="checkbox"/>	L36	(l33 or l34) and l30	11
<input type="checkbox"/>	L35	(l33 or l34) and l29	10
<input type="checkbox"/>	L34	707/100.ccls.	4707
<input type="checkbox"/>	L33	707/2-6.ccls.	12082
<input type="checkbox"/>	L32	l30 and ((query near plan) with (table or tables or record or records))	1
<input type="checkbox"/>	L31	l29 and ((query near plan) with (table or tables or record or records))	1
<input type="checkbox"/>	L30	((join near node) with ((fan-out or (fan near out)) or (inner adj1 join)))	19
<input type="checkbox"/>	L29	((join near node) near ((fan-out or (fan near out)) or (inner adj1 join)))	11
<input type="checkbox"/>	L28	l25 and ((modif\$ or updat\$ or alter\$ or chang\$) with (parent\$ or root\$ or host\$))	9
<input type="checkbox"/>	L27	((join near node) near ((left adj1 join) or (right adj1 join) or (left adj1 exception adj1 join) or (right adj1 exception adj1 join)))	3
<input type="checkbox"/>	L26	((join near node) with ((left adj1 join) or (right adj1 join) or (left adj1 exception adj1 join) or (right adj1 exception adj1 join)))	3
<input type="checkbox"/>	L25	((optimiz\$ or (query adj1 plan)) with (node\$ or tree\$ or graph\$ or hierarch\$ or level or levels or leaf or leaves or branch or branches or parent or child\$ or root) with join)	125
<input type="checkbox"/>	L24	((optimiz\$ with (query adj1 plan) with (database\$ or (data adj1 base\$)) with table\$ with (record or records) with (node\$ or leaf or leaves or branch\$ or tree\$ or level or levels or hierarch\$))	1
		<i>DB=PGPB; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L23	L22 and null	16
<input type="checkbox"/>	L22	L21 and (record or records)	18
<input type="checkbox"/>	L21	L20 not (international adj1 business adj1 machine\$.asn.	19
<input type="checkbox"/>	L20	L19 and (inner adj1 join)	25

10/487,286

<input type="checkbox"/>	L19 L18 and (child adj1 node\$)	1848
<input type="checkbox"/>	L18 (parent adj1 node\$)	2726
<input type="checkbox"/>	L17 L16 and (inner adj1 join)	2
<input type="checkbox"/>	L16 (L12 or L13) not (international adj1 business adj1 machine\$.asn.	38
<input type="checkbox"/>	L15 (query near plan).ti.	1
<input type="checkbox"/>	L14 (query adj1 plan).ti.	1
<input type="checkbox"/>	L13 (query adj1 plan).ab.	23
<input type="checkbox"/>	L12 (query adj1 plan).clm.	46
	<i>DB=USPT; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L11 L10 and (inner adj1 join)	1
<input type="checkbox"/>	L10 L9 and (join adj1 node\$)	4
<input type="checkbox"/>	L9 L8 and node\$	29
<input type="checkbox"/>	L8 (L4 or L5 or L6 or L7) not (international adj1 business adj1 machine\$.asn.	36
<input type="checkbox"/>	L7 (query adj1 plan).clm.	62
<input type="checkbox"/>	L6 (query adj1 plan).ab.	24
<input type="checkbox"/>	L5 (query adj1 plan).ti.	2
<input type="checkbox"/>	L4 (query near plan).ti.	2
<input type="checkbox"/>	L3 L2 and (inner near join)	10
<input type="checkbox"/>	L2 L1 and (join near node\$)	35

<input type="checkbox"/>	L1	299
	(5694591 5546576 6081801 6397204 6092062 5276870 6345267 5864840 6032143 6457020 5680603 5701455 5832477 5822750 6105020 6807546 5822749 5930785 6763359 6205441 6438741 5758335 5412804 5724568 5752017 6032144 6061676 6061676 5598559 6801905 6980981 6947927 5301317 5671403 6516310 5574900 6009265 6421657 5899997 6122644 5548755 6470330 5940819 6263345 5960428 5412806 6128612 5797136 5687362 5855019).pn. (6088691 6438541 6003036 5761653 5926809 6356889 6366903 6636846 6996557 5873075 6377943 5619692 5608904 5345585 6567802 6341281 6865567 5963933 5960426 6085189 6141655 5991754 6640221 6529896 5761654 6279007 6105033 6757677 6138111 6125360 6134543 5335345 5943666 5469568 5542073 5878427 6353818 6477525 6496819 5511186 6850927 5870752 5903887 6012064 6424967 6385603 5774692 5428776 5899986 6029163).pn. (6965891 5600829 5625813 5625812 6609131 5724570 6957222 5551031 5557791 6834279 5444842 6643636 6954776 6275818 5937401 5890150 6691101 5659725 5721896 5894311 5842209 6334128 6449605 6449606 6449609 6477534 6490581 6564212 6633883 6665663 6711563 6738755 6385604 5884320 5999924 5548758 5548754 6185557 6411963 6466934 6513044 5956727 6009428 5367675 7010516 6999958 6385608 6253197 6961728 4769772).pn. (5659728 5696960 5864842 5890148 6078926 5590324 5615361 5742809 5761652 5987453 6026390 5446886 5555367 5701453 5737591 5787411 5806059 5999928 6026394 6092061 6192357 6199063 6282544 6442539 6487549 6581060 6629094 6708179 6745173 6775682 6792420 5694593 5806061 5239663 6374235 6446063 6493701 4774657 5978794 6334125 5745746 6374263 6996556 6108648 6438542 6601063 6721730 6757670 6163774 6285997).pn.	

(6339768 6574639 6662175 6826562 5778355 6256621 5873074 5978793
5428774 6262984 5241648 5666525 4930072 5546571 5933417 5950185
6760718 5752028 5297280 5848408 5584024 5812840 5895465 6108651
6134546 6205451 6331983 6356896 6457000 6553371 6615206 6618720
6748392 5289460 5495608 6014656 6363371 6618719 5903893 6105017
5331637 5758145 6370524 6263331 5812996 6529901 6397214 5701454
5855012 5761657).pn. (5956705 6421663 6618729 5848406 6014655 5619688
5666526 5696961 5701460 5734887 6016488 6112197 6154747 6289336
6298338 6317738 6324533 6353828 6374252 6493708 6502089 6507835
6564204 6684203 6694306 6732096 6763350 6804678 6836777 6952692
6983291 6999967 6556988 6567806 6470287 5287493 5826257 5943665
6546402 5802521 6282281 5778354 6813617 7010521 5787410 5659754
6064820 6260037 6418448 6532458).pn.

END OF SEARCH HISTORY


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfac](#)

Terms used

database and table and record and query plan and optimization and parent and child and node and join and i

 Sort results by

 Display results
☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ Open results in a new window

 Try an [Advanced Search](#)

 Try this search in [The ACM](#)

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevan

1 [MASH: the multicasting Archie server hierarchy](#)



Adam Rosenstein, Jun Li, Si Yuan Tong

 July 1997 **ACM SIGCOMM Computer Communication Review**, Volume 27 Issue 3

Publisher: ACM Press

 Full text available: [pdf \(896.29 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The **archie**[1] system is a replicated, centralized directory server for all Anonymous FTP sites in Internet. This centralized approach has not scaled well. The **march**[2] system provides an alternative solution, using IP Multicast to distribute directory queries directly to FTP hosts. **march** suffers from redundant broadcast messages during expanding-disc search. We propose a solution to the problem of multicast flooding during expanding-disc searches by utilizing an automatic ...

2 [A case for dynamic view management](#)



Yannis Kotidis, Nick Roussopoulos

 December 2001 **ACM Transactions on Database Systems (TODS)**, Volume 26 Issue 4

Publisher: ACM Press

 Full text available: [pdf \(892.57 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Materialized aggregate views represent a set of redundant entities in a data warehouse that are used to accelerate On-Line Analytical Processing (OLAP). Due to the complex structure of the data warehouse and the different profiles of the users who submit queries, there is a need for tools that automate and ease the view selection and management processes. In this article we present DynaView, a system that manages dynamic collections of materialized aggregate views in a data warehouse.

Keywords: Data cube, OLAP, data warehousing, materialized views

3 [A framework for implementing hypothetical queries](#)



Timothy Griffin, Richard Hull

 June 1997 **ACM SIGMOD Record, Proceedings of the 1997 ACM SIGMOD international conference on Management of data SIGMOD '97**, Volume 26 Issue 2

Publisher: ACM Press

 Full text available: [pdf \(1.46 MB\)](#)



 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Previous approaches to supporting hypothetical queries have been "eager": some representation of the hypothetical state (or the corresponding delta) is materialized, and query evaluation is filtered through this representation. This paper develops a framework for evaluating hypothetical queries using a "lazy"



10/687,286

approach, or using a hybrid of eager and lazy approaches. We focus on queries having the form $\{\{U\}\&r \dots$

4 A language for queries on structure and contents of textual databases



 Gonzalo Navarro, Ricardo Baeza-Yates
 July 1995 **Proceedings of the 18th annual international ACM SIGIR conference on Research development in information retrieval**
Publisher: ACM Press
 Full text available:  [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

5 A transformation-based approach to optimizing loops in database programming languages

 Daniel F. Lieuwen, David J. DeWitt
 June 1992 **ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data SIGMOD '92**, Volume 21 Issue 2
Publisher: ACM Press
 Full text available:  [pdf\(1.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Database programming languages like O2, E, and O++ include the ability to iterate through a set of iterators can be used to express joins. This paper describes compile-time optimizations similar to transformations like join reordering for such programming constructs. This paper also shows how to use a standard transformation-based optimizer to optimize these joins. An optimizer built using the E: Opt ...



6 Abstraction-based intrusion detection in distributed environments

 Peng Ning, Sushil Jajodia, Xiaoyang Sean Wang
 November 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 4
Publisher: ACM Press
 Full text available:  [pdf\(590.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Abstraction is an important issue in intrusion detection, since it not only hides the difference between heterogeneous systems, but also allows generic intrusion-detection models. However, abstraction is an error-prone process and is not well supported in current intrusion-detection systems (IDSs). This paper presents a hierarchical model to support attack specification and event abstraction in distributed intrusion detection. The model involves three concepts: *system view*, *signature* ...

Keywords: *Cooperative information systems, heterogeneous systems, intrusion detection, misclassification detection*

7 Accelerating XPath evaluation in any RDBMS

 Torsten Grust, Maurice Van Keulen, Jens Teubner
 March 2004 **ACM Transactions on Database Systems (TODS)**, Volume 29 Issue 1
Publisher: ACM Press
 Full text available:  [pdf\(781.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This article is a proposal for a database index structure, the *XPath accelerator*, that has been specifically designed to support the evaluation of XPath path expressions. As such, the index is capable to support all XPath axes (including ancestor, following, preceding-sibling, descendant-or-self, etc.). This feature makes the index stand out among related work on XML indexing structures which had a focus on the child and descendant axes only. The index has been designed with a close ...

Keywords: Main-memory databases, XML, XML indexing, XPath

8 Active database systems

Norman W. Paton, Oscar Díaz

March 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 1

Publisher: ACM Press

Full text available: pdf(2.68 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Active database systems support mechanisms that enable them to respond automatically to events taking place either inside or outside the database system itself. Considerable effort has been directed towards improving understanding of such systems in recent years, and many different proposals have been made and applications suggested. This high level of activity has not yielded a single agreed-upon approach to the integration of active functionality with conventional database systems.

Keywords: active databases, events, object-oriented databases, relational databases

9 Adaptive multi-stage distance join processing

Hyoseop Shin, Bongki Moon, Sukho Lee

May 2000 **ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD '00**, Volume 29 Issue 2

Publisher: ACM Press

Full text available: pdf(333.42 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A spatial distance join is a relatively new type of operation introduced for spatial and multimedia applications. Additional requirements for ranking and stopping cardinality are often combined with spatial distance join in on-line query processing or internet search environments. These requirements present new challenges as well as opportunities for more efficient processing of spatial distance join queries. In this paper, we first present an efficient k -distance join algorithm.

10 Advanced SQL modeling in RDBMS

Andrew Witkowski, Srikanth Bellamkonda, Tolga Bozkaya, Nathan Folkert, Abhinav Gupta, John Han Sheng, Sankar Subramanian

March 2005 **ACM Transactions on Database Systems (TODS)**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: pdf(279.06 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Commercial relational database systems lack support for complex business modeling. ANSI SQL treats relations as multidimensional arrays and defines multiple, interrelated formulas over them, which are needed for business modeling. Relational OLAP (ROLAP) applications have to perform complex queries using joins, SQL Window Functions, complex CASE expressions, and the GROUP BY operator in pivot operation. The designated place in SQL for calculations is the SELECT clause, which is not suitable for complex business modeling.

Keywords: Excel, OLAP, analytic computations, spreadsheet

11 An adaptive query execution system for data integration

Zachary G. Ives, Daniela Florescu, Marc Friedman, Alon Levy, Daniel S. Weld

June 1999 **ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international conference on Management of data SIGMOD '99**, Volume 28 Issue 2

Publisher: ACM Press

Full text available: pdf(1.59 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Query processing in data integration occurs over network-bound, autonomous data sources. This paper presents extensions to traditional optimization and execution techniques for three reasons: there is an increasing demand for quality statistics about the data, data transfer rates are unpredictable and bursty, and slow or unreliable data sources can often be replaced by overlapping or mirrored sources. This paper presents the design of a data integration system, designed to support adaptivity at its core using a query optimizer.

12 An overview of query optimization in relational systems



Surajit Chaudhuri

May 1998

Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on of database systems

Publisher: ACM Press

Full text available: pdf(1.42 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

13 An XML query engine for network-bound data

Zachary G. Ives, A. Y. Halevy, D. S. Weld

December 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, v 4

Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(351.86 KB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

XML has become the lingua franca for data exchange and integration across administrative and boundaries. Nearly all data providers are adding XML import or export capabilities, and standard Schemas and DTDs are being promoted for all types of data sharing. The ubiquity of XML has reduced the major obstacles to integrating data from widely disparate sources - namely, the heterogeneous formats. However, general-purpose integration of data across the wide area is also required ...

Keywords: Data integration, Data streams, Query processing, Web and databases, XML

14 Anatomy of a native XML base management system

T. Fiebig, S. Helmer, C.-C. Kanne, G. Moerkotte, J. Neumann, R. Schiele, T. Westmann

December 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, v 4

Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(300.97 KB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Several alternatives to manage large XML document collections exist, ranging from file systems to relational or other database systems to specifically tailored XML base management systems. In this paper we give a tour of Natix, a database management system designed from scratch for storing and managing XML data. Contrary to the common belief that management of XML data is just another application, traditional databases like relational systems, we illustrate how almost every component in a ...

Keywords: Database, XML

15 Answering complex SQL queries using automatic summary tables



Markos Zaharioudakis, Roberta Cochrane, George Lapis, Hamid Pirahesh, Monica Urata

May 2000

ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD '00, Volume 29 Issue 2

Publisher: ACM Press

Full text available: pdf(185.85 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We investigate the problem of using materialized views to answer SQL queries. We focus on materialized decision-support queries, which involve joins, arithmetic operations and other (possibly user-defined) functions, aggregation (often along multiple dimensions), and nested subqueries. Given the complexity of such queries, the vast amounts of data upon which they operate, and the requirement for interactive response times, the use of materialized views (MVs) of similar complexity is often mandatory ...


16 Articles: Bringing order to query optimization



Giedrius Slivinskas, Christian S. Jensen, Richard Thomas Snodgrass

June 2002

ACM SIGMOD Record, Volume 31 Issue 2

Publisher: ACM PressFull text available:  [pdf\(1.12 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#)

A variety of developments combine to highlight the need for respecting order when manipulating data. For example, new functionality is being added to SQL to support OLAP-style querying in which ordering is frequently an important aspect. The set- or multiset-based frameworks for query optimization that are currently being taught to database students are increasingly inadequate. This paper presents a framework for query optimization that extends existing frameworks to also capture ordering. A 1 ...

17 [Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining](#)



Robbert Van Renesse, Kenneth P. Birman, Werner Vogels

May 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 2**Publisher:** ACM PressFull text available:  [pdf\(341.62 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Scalable management and self-organizational capabilities are emerging as central requirements for the generation of large-scale, highly dynamic, distributed applications. We have developed an entire distributed information management system called Astrolabe. Astrolabe collects large-scale system data, permitting rapid updates and providing on-the-fly attribute aggregation. This latter capability permits an application to locate a resource, and also offers a scalable way to track system state ...

Keywords: Aggregation, epidemic protocols, failure detection, gossip, membership, publish-subscribe, scalability

18 [Authentication: Redundancy and information leakage in fine-grained access control](#)



Govind Kabra, Ravishankar Ramamurthy, S. Sudarshan

June 2006 **Proceedings of the 2006 ACM SIGMOD international conference on Management of Data (SIGMOD '06)****Publisher:** ACM PressFull text available:  [pdf\(340.16 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The current SQL standard for access control is coarse grained, in that it grants access to all rows or none. Fine-grained access control, which allows control of access at the granularity of individual rows and to specific columns within those rows, is required in practically all database applications. There are several models for fine grained access control, but the majority of them follow a view replacement model. There are two significant problems with most implementations of this model ...

Keywords: fine-grained access control, information leakage, query optimization, redundancy

19 [Automatic high-quality reengineering of database programs by abstraction, transformation, and reimplementation](#)



Yossi Cohen, Yishai A. Feldman

July 2003 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 12 Issue 3**Publisher:** ACM PressFull text available:  [pdf\(245.97 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Old-generation database models, such as the indexed-sequential, hierarchical, or network models, require record-level access to their data, with all application logic residing in the hosting program. In contrast, modern relational databases can perform complex operations, such as filter, aggregation, and join, on large numbers of records without an external specification of the record-access logic. Programs written for relational databases attempt to move as much of the application logic as possible into the database ...

Keywords: Database program reengineering, query graphs, temporal abstraction, the plan calculator

20 Avoiding Cartesian products for multiple joins



Shinichi Morishita

January 1997 **Journal of the ACM (JACM)**, Volume 44 Issue 1

Publisher: ACM Press

Full text available: [pdf\(583.81 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [rev](#)

Computing the natural join of a set of relations is an important operation in relational database. The ordering of joins determines to a large extent the computation time of the join. Since the number of possible orderings could be very large, query optimizers first reduce the search space by using heuristics and then try to select an optimal ordering of joins. Avoiding Cartesian products is a common heuristic for reducing the search space, but it cannot guarantee optimal or ...

Keywords: Cartesian product, database scheme, join expression tree, join strategy, optimality

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)


[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) |

Welcome United States Patent and Trademark Office

Search Results

[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Results for "((inner join<in>metadata) <and> (query<in>metadata))"

e-mail

Your search matched 1 of 488547 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

» Search Options

[View Session History](#)[New Search](#)

Modify Search

((inner join<in>metadata) <and> (query<in>metadata))

Search

☐ Check to search only within this results setDisplay Format: ☒ Citation ☐ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

view selected items

[Select All](#) [Deselect All](#)

- ☐ 1. Outer joins and filters for Instantiating objects from relational databases
 Byung Suk Lee; Wiederhold, G.;
Knowledge and Data Engineering, IEEE Transactions on
 Volume 6, Issue 1, Feb. 1994 Page(s):108 - 119
 Digital Object Identifier 10.1109/69.273031
[AbstractPlus](#) | Full Text: [PDF\(1056 KB\)](#) IEEE JNL
[Rights and Permissions](#)

 Indexed by
 Inspec
[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

10/687,286